

SYSTEM AND METHOD FOR TESTING AN INTERCONNECT IN A COMPUTER SYSTEM

Background

5 Computer systems generally include a number of components that are electrically connected to one another. These components include one or more processors, memory devices, input / output (I/O) devices, and controllers for the memory and I/O devices. Because the components can be contained in various types of housing, the connections between the components can take numerous forms. For example, components may be microchips that may plug into or be soldered into slots sockets on a motherboard. Components may also take the form of circuit boards that have edge connectors that plug into slots on the motherboard. In addition, components may be connected to a computer system using cables that connect components to connectors in the motherboard or into
10
15 plugs in the chassis that houses the motherboard.

Regardless of the connection mechanism between two components in a computer system, a failure in a connection between two or more components may cause broader failures to occur in the system and possibly cause the system to crash. Although some diagnostic testing of interconnections between components may occur in response to a computer system being turned on or reset, this type of testing may not detect failures in computer systems that are left on and not reset for extended periods of time. In addition, certain failures of an interconnect may not appear without rigorous pattern testing of the interconnect. Interconnect failures that occur in computer systems during operation may not be
20
25 detected until they cause undesirable results such as a crash.

Accordingly, it would be desirable to be able to detect interconnect failures between components in a computer system before the failures cause undesirable results during operation of the system.

Summary

30 According to one exemplary embodiment, a computer system is provided that includes an operating system, a first component that comprises a first test

module, a second component that comprises a second test module, and an interconnect coupling the first component and the second component. The first test module is configured to provide a first test pattern to the second test module on the interconnect in response to a first signal from the operating system.

5

Brief Description of the Drawings

Figure 1 is a block diagram illustrating an embodiment of a computer system that includes interconnect test modules.

Figure 2 is a block diagram illustrating an embodiment of an interconnect test module.

10

Figure 3a is a block diagram illustrating an embodiment of selected portions of the computer system shown in Figure 1.

Figure 3b is a block diagram illustrating an embodiment of selected portions of the computer system shown in Figure 1.

15

Figure 4 is a flow chart illustrating an embodiment of a method for testing an interconnect during operation of a computer system.

Detailed Description

In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural or logical changes may be made without departing from the scope of the present invention. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims.

25

In one aspect of the present disclosure, a computer system includes interconnect test modules configured to perform tests on an interconnect in the computer system. The interconnect test modules are included in components that are coupled to an interconnect. To test the interconnect, an interconnect test module causes the components coupled to the interconnect to be de-allocated from use by the computer system and then provides signals in the form of test

30

patterns to a second interconnect test module. The second interconnect test module detects errors in response to receiving the test patterns. The second interconnect test module may also provide test patterns back to the first interconnect test module. If an error is detected in the interconnect, remedial
5 action is performed.

Figure 1 is a block diagram illustrating an embodiment of a computer system 100 that includes interconnect test modules 150a, 150b, 150c, and 150d. As used herein, 'interconnect test module 150' refers to any one of interconnect test modules 150a, 150b, 150c, or 150d, and 'interconnect test modules 150'
10 refers to the set of interconnect test modules 150a, 150b, 150c, and 150d.

Computer system 100 may be any type of computer system such as a handheld, desktop, notebook, mobile, workstation, or server computer. Computer system 100 includes processors 110a through 110(n), a core electronics complex 120, a memory 130, and a set of input / output (I/O) devices
15 140. Processors 110a through 110(n) are each coupled to core electronics complex 120 using a set of bus connections 142. Bus connections 142 comprise a set of system busses. Core electronics complex 120 is coupled to memory 130 and I/O devices 140 using connections 144 and 146, respectively. Core electronics complex 120 may also be referred to as a chipset.

20 Computer system 110a includes any number of processors 110 greater than or equal to one. As used herein, 'processor 110' refers to any one of processors 110a through 110(n), and 'processors 110' refers to the set of processors 110a through 110(n).

Processor 110a is coupled to a cache 112, and processor 110b includes a
25 cache 114. Caches 112 and 114 may store any type of information such as instructions and data. Other processors 110 may include or be operable with any type or number of caches.

Computer system 100 also includes an operating system 132 that is executable by one or more of processors 110. In response to being turned on or
30 reset, one or more of processors 110 cause operating system 132 to be booted and executed. Processors 110 execute instructions from operating system 132 and other programs using memory 130.

Core electronics complex 120 includes a system controller 122 and a set of I/O controllers 124. System controller 122 includes a memory controller 126 which is configured to store information into and read information from memory 130 in response to write and read transactions, respectively, from processors 110
5 and I/O devices 140. Memory controller 126 may include hardware and / or software configured to perform memory scrubbing or other error correction functions on memory 130 in response to reading information from memory 130.

I/O controllers 124 may include any type and number of controllers configured to manage one or more I/O devices 140. Examples of I/O controllers
10 124 include IDE/ATA controllers, SATA controllers, PCI controllers, SCSI controllers, USB controllers, IEEE 1394 (Firewire) controllers, PCMCIA controllers, parallel port controllers, and serial port controllers. In one embodiment, I/O controllers 124 comprise multiple microchips that include an intermediate bus coupled to system controller 122, PCI controllers coupled to the
15 intermediate bus, and SCSI, IDE and others controllers coupled to the PCI controllers. As used herein, 'I/O controller 124' refers to a single I/O controller in I/O controllers 124, and 'I/O controllers 124' refers to the set of I/O controllers 124.

Memory 130 comprises any type of memory managed by memory
20 controller 126 such as RAM, SRAM, DRAM, SDRAM, and DDR SDRAM. In response to commands from system firmware (not shown) or operating system 132, memory controller 130 may cause information to be loaded from an I/O device 140 such as a hard drive or a CD-ROM drive into memory 130.

I/O devices 140 may include any type and number of devices configured
25 to communicate with computer system 100 using I/O controllers 124. Each I/O device 140 may be internal or external to computer system 100 and may couple to an expansion slot in a motherboard or a connector in a chassis that houses computer system 100 that is in turn coupled to an I/O controller 124. I/O devices 140 may include a network device (not shown) configured to allow
30 computer system 100 to communicate with other computer systems and a storage device (not shown) configured to store information. As used herein, 'I/O device

140' refers to a single I/O device in I/O devices 140, and 'I/O devices 140' refers to the set of I/O devices 140.

Interconnect test modules 150 operate to perform tests on an interconnect between two components of computer system 100 during operation, i.e.,
5 subsequent to operating system 132 being booted, of computer system 100. In the embodiment shown in Figure 1, interconnect test modules 150a and 150b are configured to perform tests on the interconnect, i.e., system bus 142, between processor 110a and system controller 122. Similarly, interconnect test modules 150c and 150d are configured to perform tests on the interconnect, i.e.,
10 connection 146, between an I/O controller 124 and an I/O device 140.

Figure 2 is a block diagram illustrating an embodiment of interconnect test module 150. Interconnect test module 150 includes a test unit 202, a switching mechanism 204, and an error log 206. Switching mechanism 204 is coupled to a connection 212 from a component and a test connection 214 from
15 test unit 202. Switching mechanism 204 can either be an explicit switching component in the ASIC or the functionality could be built into each of the individual ASIC I/O pads.

Test unit 202 provides a control signal 216 to switching mechanism 204 to cause either connection 212 or test connection 214 to be coupled to a
20 connection 218. Switching mechanism 204 couples connection 218 to either connection 212 or test connection 214 in response to receiving control signal 216 from test unit 202. During normal operation, test unit 202 causes switching mechanism 204 to couple connection 212 to connection 218. In a test mode of operation, however, test unit 202 causes switching mechanism 204 to couple
25 connection 214 to connection 218 to allow test unit 202 to perform tests on an interconnect coupled to connection 218.

Test unit 202 comprises a state machine configured to cause tests to be performed on an interconnect coupled to connection 218. The tests may include any suitable combinations of signals, referred to as test patterns, provided to and
30 conveyed on an interconnect coupled to connection 218. A second interconnect test module 150 coupled to the interconnect receives the test patterns to determine whether an error occurred in transmitting the test patterns from the

first interconnect test module 150 to the second interconnect test module 150. If an error occurs, the second interconnect test module 150 notes the failure in its error log 206. The second interconnect test module 150 also provides test patterns to the first interconnect test module 150. In response to receiving test
5 patterns from the second interconnect test module 150, test unit 202 in the first interconnect test module 150 detects any errors and records the errors in the error log 206 of the first interconnect test module 150.

To perform tests on an interconnect, test unit 202 first causes computer system 100 to de-allocate components coupled to the interconnect from use by
10 operating system 132. In one embodiment, test unit 202 accomplishes this task by sending a signal in the form of a request to operating system 132. Operating system 132 de-allocates the components and may provide a signal to test unit 202 to indicate that the components have been de-allocated. Operating system 132 may also allocate other components to at least temporarily replace those that
15 have been de-allocated. In other embodiments, test unit 202 may cause components to be de-allocated in other ways.

After the components coupled to the interconnect have been de-allocated, test unit 202 causes tests to be performed on the interconnect in conjunction with a second test unit 202. Subsequent to or while performing the tests, test unit 202
20 causes operating system 132 to be notified of any errors and that the tests are complete. If any errors occur, operating system 132 may cause appropriate remedial action to be performed such as keeping the interconnect and the components coupled to it offline and notifying a system administrator. If no errors occur, operating system 132 may cause the components coupled to the
25 interconnect to be re-allocated for use in computer system 100.

Figures 3a and 3b are block diagrams illustrating embodiments of selected portions of the computer system shown in Figure 1. In particular, Figures 3a and 3b illustrate two possible uses of interconnect test modules 150. Many other uses of interconnect test modules 150 are possible and contemplated.

30 In Figure 3a, processor 110a, which includes interconnect test module 150a, is coupled to system controller 122, which includes interconnect test module 150b, through a connector 300. Connector 300 may be a socket, slot, or

other connection mechanism on the motherboard (not shown) of computer system 100. Switching mechanism 204a is connected to a chip core 302 of processor 100a and a system bus 142a which is connected through connector 300 to switching mechanism 204b. Switching mechanism 204b is also connected to
5 a chip core 304 of system controller 122.

During normal operation of processor 110a and system controller 122, test units 202a and 202b cause switching mechanisms 204a and 204b, respectively, to connect chip cores 302 and 304, respectively, to system bus 142a.

10 To test the interconnect between processor 110a and system controller 122, i.e., system bus 142a through connector 300, test units 202a and 202b cause switching mechanisms 204a and 204b, respectively, to connect test units 202a and 202b, respectively, to system bus 142a. Test unit 202a and / or 202b initiate testing of the interconnect by causing processor 110a and system bus 142a to be
15 de-allocated from use by operating system 132 using operating system 132. An un-allocated processor 110 may be allocated for use in place of processor 110a during the tests.

During testing, test unit 202a generates test patterns and provides the test patterns to test unit 202b. Test unit 202b receives the test patterns and compares
20 the received test patterns expected test patterns to determine if an error occurred. Test unit 202b stores errors that it detects in error log 206b. Similarly, test unit 202b generates test patterns and provides the test patterns to test unit 202a. Test unit 202a receives the test patterns and compares the received test patterns expected test patterns to determine if an error occurred. Test unit 202a stores
25 errors that it detects in error log 206a. In addition to the test patterns, test units 202a and 202b may also generate and send signals to communicate with one another. Further, test units 202a and 202b may generate and send signals to operating system 132 to report errors and other test results.

Although test module 150b is shown in system controller 122 in the
30 embodiments of Figures 1 and 3a, test module 150b may be included in memory controller 126 in other embodiments.

In Figure 3b, I/O controller 124a, which includes interconnect test module 150c, is coupled to I/O device 140a, which includes interconnect test module 150d, through a connector 320. Connector 320 may be a socket, slot, or other connection mechanism on the motherboard (not shown) of computer system 100. Switching mechanism 204c is connected to a chip core 312 of I/O controller 124a and a bus 146a which is connected through connector 320 to switching mechanism 204d. Switching mechanism 204d is also connected to a chip core 314 of I/O device 140a.

During normal operation of I/O controller 124a and I/O device 140a, test units 202c and 202d cause switching mechanisms 204c and 204d, respectively, to connect chip cores 312 and 314, respectively, to bus 146.

To test the interconnect between I/O controller 124a and I/O device 140a, i.e., bus 146a through connector 320, test units 202c and 202d cause switching mechanisms 204c and 204d, respectively, to connect test units 202c and 202d, respectively, to bus 146a. Test unit 202c and / or 202d initiate testing of the interconnect by causing I/O device 140a and system bus 146a to be de-allocated from use by operating system 132 using operating system 132.

During testing, test unit 202c generates test patterns and provides the test patterns to test unit 202d. Test unit 202d receives the test patterns and compares the received test patterns expected test patterns to determine if an error occurred. Test unit 202d stores errors that it detects in error log 206d. Similarly, test unit 202d generates test patterns and provides the test patterns to test unit 202c. Test unit 202c receives the test patterns and compares the received test patterns expected test patterns to determine if an error occurred. Test unit 202c stores errors that it detects in error log 206c. In addition to the test patterns, test units 202c and 202d may also generate and send signals to communicate with one another. Further, test units 202c and 202d may generate and send signals to operating system 132 to report errors and other test results.

Figure 4 is a flow chart illustrating an embodiment of a method for testing an interconnect during operation of computer system 100 using interconnect test module 150. An interconnect test is initiated by operating system 132 or interconnect test module 150 as indicated in a block 400.

Interconnect tests may be scheduled periodically and may be scheduled in response to selections made by user interacting with operating system 132.

One or more components coupled to the interconnect are de-allocated from use by operating system 132 as indicated in a block 402. As noted above,
5 interconnect test module 150 may send a request or other signal to operating system 132 to cause the component or components to be de-allocated. Operating system 132 may respond by providing a signal back to interconnect test module 150 to indicate that the component(s) have been de-allocated, i.e., that the interconnect is available for testing by interconnect test module 150. Substitute
10 components are allocated to replace the de-allocated components, if available, as indicated in a block 404.

Tests are performed on the interconnect by interconnect test module 150 as indicated in a block 406. Interconnect test module 150 performs tests by generating test patterns and providing the test patterns across the interconnect to
15 a second interconnect test module 150 as shown in Figures 1, 3a, and 3b. A determination is made as to whether an error has been detected on the interconnect by interconnect test module 150 as indicated in a block 408. To detect an error, the second interconnect test module 150 compares received test patterns to expected test patterns. If an error has been detected on the
20 interconnect, then remedial action, such as notifying operating system 132 and the other interconnect test module 150, taking the components coupled to the interconnect offline, and notifying a system administrator, is performed as indicated in a block 410.

If no error has been detected on the interconnect, then a determination is
25 made as to whether there are more tests to perform on the interconnect as indicated in a block 412. If there are more tests to be performed on the interconnect, then the function of block 406 is repeated as indicated. If there are no more tests to be performed on the interconnect, then results are reported to operating system 132 by interconnect test module 150 as indicated in a block
30 414. The components coupled to the interconnect are re-allocated as indicated in a block 416.

In the embodiments described herein, interconnect test module 150 and the components therein may comprise hardware, software, or any combination of hardware and software.

Although specific embodiments have been illustrated and described
5 herein, it will be appreciated by those of ordinary skill in the art that a variety of alternate and/or equivalent implementations may be substituted for the specific embodiments shown and described without departing from the scope of the present invention. This application is intended to cover any adaptations or variations of the specific embodiments discussed herein. Therefore, it is
10 intended that this invention be limited only by the claims and the equivalents thereof.